Associative Fine-Tuning of Biologically Inspired Active Neuro-Associative Knowledge Graphs

Adrian Horzyk
Dept. Automatics and Biomedical Engineering
AGH University of Science and Technology
Krakow, Poland
horzyk@agh.edu.pl

Janusz A. Starzyk
University of Information Technology and Management in Rzeszow
and School of EECS, Ohio University
Rzeszow, Poland, and Athens, OH 45701.J., USA
starzykj@ohio.edu

***Abstract* — This paper introduces a new tuning algorithm that improves the associative training algorithm of active neuro-associative knowledge graphs (ANAKG). We also expand the definition of synaptic weights, using new multiplicative factors. Biological neural networks are sparse and developed in neuronal plasticity processes adapting them to the repeatable combinations of the input stimuli. Real neurons connect conditionally according to neural activity and on demand of some biochemical processes. They do not connect to all neurons in subsequent layers as is usually performed in artificial neural networks. For more than a decade, scientists conducted extensive research on adaptation mechanisms which use sparsely connected neural structures that can specialize and adapt faster to training data. This approach is also widely used in various learning strategies of deep neural networks. Conditional creation of sparse connections and fine-tuning of their weights in complex associative neuronal ANAKG structures are the main contributions of this paper. The significant improvement of recalling of the context-based associations was verified experimentally.**

Keywords — brain-inspired associative learning, spiking neural networks, deep learning, tuning of weights, sequential patterns, associative representation of knowledge, active neuro-associative knowledge graphs, hetero-associative memory.

I. Introduction

In computational intelligence, there are many different approaches to learn sequential patterns. We distinguish linear models like Hidden Markov Models or linear autoencoders, and non-linear models, e.g. recurrent neural networks (RNN), long-short term memories (LSTM), finite state machines (FSM), long-term memories (LTM), associative memories (AM), and various deep learning approaches [4], [5], [7], [14], [17], [18]. Many approaches require to construct large structures and use a lot of computing power to construct a computational model and memorize sequential patterns. These are very common and useful in practical applications like video or image recognition, language understanding, and signal processing. The ability to adapt to sequential patterns is also required to build cognitive systems [1], [2], [15], [16] which should be able to represent various scenarios, actions, and associations between objects.

This paper introduces a new algorithm for tuning the synaptic weights of ANAKG networks. This improves the associative learning algorithm introduced in [6] and used to create semantic memories in [7]. The ANAKG neural networks are constructed from receptors that are sensitive to input data, associative spiking neurons (ASN) which work in time and distribute spikes in complex graph structures, and effectors that transform spikes to output data. The ANAKG networks can be adapted to learn many sequences that are combined in such a way that all instances of the same elements are represented by the same ASN neurons. Thanks to these aggregations, various sequences partially share the same neurons and semantically bind them together creating a kind of semantic memories. In the ANAKG networks, these special associative neurons are automatically connected in the graph structure, reproducing the training sequences. Connections are created not only to the neurons representing direct predecessors (previous elements) but also to the neurons representing previously learned sequence elements. Such connections are necessary to establish the appropriate contexts that allow differentiation between the successors that should be executed next. Therefore, such aggregated representations of elements are convenient and required from the semantic representation, compression of information, and generalization point of view. However, they cause essential difficulties when recalling training sequences which are very correlated in their first elements that represent recalling contexts. Hence, it is necessary to adapt and fine-tune the connection weights of such complex networks to avoid unintended activations and to trigger activations of desirable neurons in the appropriate order defined by the training sequences.

The solution to this problem, described in [7], laid the solid foundation for such associative representations, but the recalled sequences were sometimes shortened or recalled with extra activations of unintended neurons. Thus, the memories of past events were not very accurate. The solution presented in this paper proposes a new algorithm which allows for the fine-tuning of synaptic weights. Obtained new weights avoid all undesired activations, and does not cut recalled sequences if the recalling context is unique and elements are not repeated. The associative training algorithm presented in [7] was able to quickly create and develop an ANAKG network from scratch for any given set of sequences and compute so-called synaptic efficacies and permeabilities which were used to determine synaptic weights. The synaptic efficacies were based on the synaptic activity associated with the activity of the postsynaptic neurons, taking into account the number of stimulations and activations of the postsynaptic neurons related to these stimulations in time.

978-1-5386-9276-9/18/$31.00

The fine-tuning algorithm introduced in this paper uses time as an important computational factor which determines final strength of weights and adaptive abilities of these kinds of neural networks. We assume that all neuronal processes run in time and that neurons can relax at different rates dependently on the time interval from their last stimulations. The relaxation process makes neurons returning to their resting states gradually. Suitably adjusted weights should allow differentiating input stimulations that should not activate the neuron from those that should do it to forward activation to other connected neurons. The main difficulty of training such networks is that small differences in time and strength of stimulations result in big differences in recalled associations. Neurons representing elements from a given sequence should be activated in proper time, i.e. neither premature nor too late. Hence, it is very important to stabilize such training process and set appropriate and precisely tuned weights to be able to charge and trigger subsequent neurons in the required order defined by any set of training sequences.

II. Associative Spiking Neurons and ANAKG Networks

Models of neurons have evolved in time from the most simple like McCulloch-Pitts neurons [4], through various models using continuous and non-linear activation functions [4] to more biologically plausible spiking models [3], [8], [11], [13]. Spiking neurons try to model biological neurons and the processes that take place inside them. From the computational point of view, it is unnecessary to reproduce biochemical processes faithfully but to model the functionality of real neurons appropriately. On this basis, a new model of neurons, called associative spiking neurons (ASN), was introduced and described in [6], [7], [10] and used in the ANAKG networks.

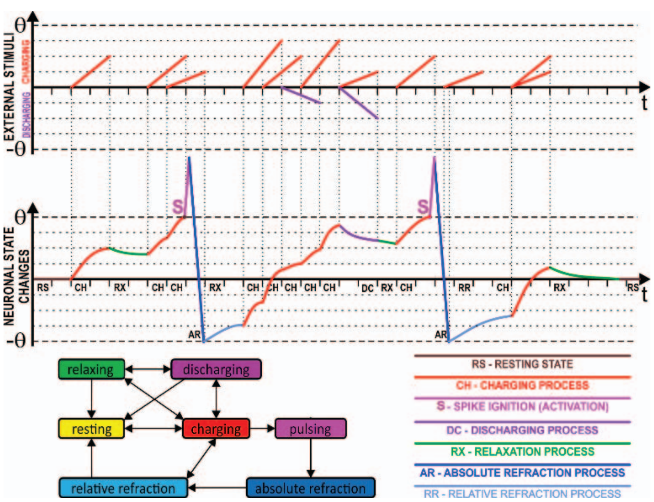

Fig. 1. The ASN internal state changes according to sample input stimuli influencing the neuronal state in time.

ASN neurons implement the time approach to neural computations and model characteristic processes of real neurons, i.e. charging (for excitation stimuli), discharging (for inhibition stimuli), relaxation (which takes place after the stimuli which have not activated neurons), absolute refraction (which is automatically started after neuron activations), and relative refraction (which starts when absolute refraction is finished) (Fig. 1). In addition to these processes, each ASN neuron can also be in the resting state. ASN neurons charged to the activation threshold level are activated and produce a spike, changing their internally running process from charging to absolute refraction. They use a different scale to model these processes and internal states than biological neurons [12] and other spiking models of neurons. ASN neurons operate in range $(0, \theta)$ for all excited states (e.g. during relaxation) as a result of charging processes executed for excitation stimuli, in range $[-\theta, \theta]$ to model absolute refraction, in range $(-\theta, 0)$ to model a relative refraction state, where θ is the activation threshold of the neuron. ASN neurons take the state value 0 when neurons are resting, and take the value θ or higher when neurons spike.

The ASN neurons have built-in plasticity rules which allow them to connect conditionally when the duration between their activations is no longer than the given maximum duration of plasticity d_{plast}^{max}. It is usually set to the time interval equal relaxation time of the neuron from its under-threshold excitation level. The plasticity process is not initiated when the time that lapsed between the activations of two neurons is less than the given minimum duration of plasticity d_{plast}^{min} that is typically set to the time interval equal absolute refraction period. When the duration between the activations of two neurons is less than d_{plast}^{min}, the activations are treated as simultaneous. When the duration d between the activations of two neurons is in the range of $\left[d_{plast}^{min}, d_{plast}^{max}\right]$, then the earlier activated neuron is connected to the later activated neuron and the weight of this connection is computed and fine-tuned as described further in this paper.

Assume that we have a training sequence set $\mathbb{S} = \{S^1, \ldots, S^N\}$ where $S^n = \left[E_1^n, \ldots, E_m^n, \ldots, E_{m+r}^n, \ldots, E_{K_n}^n\right]$ is a sequence of K_n elements. The synaptic efficacy is computed for each two connected neurons N_m and N_{m+r} representing two elements E_m^n and E_{m+r}^n of each training sequence S^n which contains them. The same elements E_m^n and E_{m+r}^n can occur in different training sequences. If the sequence S^n contains many elements $K_n > K^{max}$, then the connections are only created between subsequent elements when $r \leq K^{max}$. They can be separated by a number of other objects $(r-1)$, where $r \geq 1$. The time differences between observations of E_m^n and E_{m+r}^n objects represented by the activations of neurons N_m and N_{m+r} affect the computation of components of the sum (1) defining the **synaptic efficacy**:

$$\delta_{N_m,N_{m+r}} = \sum_{\{(S_m,S_{m+r})\in S^n\in\mathbb{S}\}} 1\Big/\left(1+\frac{\Delta t^A-\Delta t^C}{\theta_{N_{m+r}}\cdot\Delta t^R}\right)^{\tau} \quad (1)$$

where

- Δt^A is the period of time that lapsed between the stimulation of the synapse between the N_m and N_{m+r} neurons and the activation of the postsynaptic neuron N_{m+r} during training (note that not every synapse stimulation leads promptly to the activation of the postsynaptic neuron),
- Δt^C is the period of time necessary to charge and activate the postsynaptic neuron N_{m+r} after stimulating the synapse

between the N_m and N_{m+r} neurons (here $\Delta t^C = 30$ms),

Δt^R is the maximum period of the time during which the postsynaptic neuron N_{m+r} recovers and returns to its resting state after its charging that was not strong enough to activate this neuron (here $\Delta t^R = 200$ms),

$\theta_{N^n_{m+r}}$ is the activation threshold of the postsynaptic neuron N_{m+r} (here $\theta_{N^n_{m+r}} = 1$),

τ is a context influence factor changing the influence of the previously activated and connected neurons on the postsynaptic neuron N_{m+r} (here $\tau = 4$).

The synaptic efficacy (1) for each synapse takes into account all training sequences that contain the time-ordered succession of elements E^n_m and E^n_{m+r}. The synaptic connection between the neurons representing elements E^n_m and E^n_{m+r} is denoted here as $N_m \rightarrow N_{m+r}$. In practice, the synaptic efficacy measures how strong an input stimulation influences the postsynaptic neuron activity due to the elapsed time between the activations of pre- and postsynaptic neurons. Synaptic efficacy weights and integrates all related activities of the connected neurons during training.

Active neuro-associative knowledge graphs (ANAKG) are devoted to the consolidation of training sequences of objects represented by ASN neurons. ANAKG networks conditionally connect neurons, quickly calculate synaptic efficiencies and permeabilities [7], and fine-tune their weights as described in the next section. They also consist of receptors that supply neurons with sequential combinations of input stimuli and effectors that are used as output interfaces that affect actuators or send output data to other applications as described in [6] and [7].

Each ANAKG network integrates representations of many training sequences which are typically correlated in such a way that the repeated sequence elements are represented by the same neurons. It means that the representations of the same elements defining various training sequences are not duplicated. Such neurons bind and relate training sequences together and thanks to the contextual connections between their elements, it is possible to recall these sequences using unique initial contexts [7], [10]. When a new or non-unique context is presented, then the ANAKG network retrieves the most frequent training sequence, the strongest subsequence, or creates a new sequence constructed from some parts of the trained sequences.

The synaptic efficacy (1) contains the influence of the presynaptic neuron N_m on the postsynaptic neuron N_{m+r} activity. After the presynaptic neuron N_m activity was observed, the **synaptic permeability** of the connection $N_m \rightarrow N_{m+r}$ is computed using updated synaptic efficacy and one of the following formulas:

$$p = \theta \cdot \frac{2 \cdot \delta}{\eta + \delta} \tag{2}$$

$$p = \theta \cdot \frac{\eta \cdot \delta}{\eta \cdot \delta + \eta^2 - \delta^2} \tag{3}$$

where

η is a number of activations of the presynaptic neuron N_m during training of the training sequence set $\mathbb{S}$,

δ is a synaptic efficacy computed for this synapse.

In this paper, the synaptic weights based on permeabilities are further adjusted in the tuning process. This process can conditionally strengthen or weaken previously computed synaptic permeabilities of all synapses in the ANAKG network using multiplication factors m > 0. These factors model multiple connections between the same real neurons in order to achieve better-suited associations reproducing training sequences. Moreover, each neuron can influence the postsynaptic neuron in one of two ways using excitatory (c = 1) or inhibitory (c = −1) connections. This paper uses synaptic weights of ASN neurons which takes into account three factors (4):

$$w = c \cdot p \cdot m \tag{4}$$

The permeability p and multiplication m factors are computed separately. Permeabilities are computed during the construction process of the ANAKG networks as in [7], while multiplication factors are calculated and updated during the fine-tuning process described in the section III. Fine-tuning repeatedly checks whether the training sequences are already correctly recalled from their first input elements. If the tuning process is only partially successful, it is continued until the goal is achieved or it is detected that the perfect tuning is not possible. The lack of possibility to tune weights perfectly comes from the non-uniqueness of the first elements of many training sequences.

III. Associative Tuning of ANAKG Networks

In biology, living creatures learn on the basis of incoming stimuli. The associative training algorithm of ANAKG networks assumes that there are no other training stimuli than those coming from repetitions of training sequences, like presenting words sequentially as organized in these training sequences. Thus, each training sequence S^n defined by subsequent elements $E^n_1, E^n_2, \ldots, E^n_K$ is presented to the ANAKG network in successive moments of time to improve the associations between its memory elements. During the associative training, we assume that the network organizes its structure only through presentation of input data without any information coming from a supervisor and without pointing any labels of classes or desired output values in regression tasks. Class labels or any desired output values (e.g. class labels) can be presented on the network inputs in a similar way as other data without assigning them an extra role in the adaptation process. The network self-organizes making associations between all its inputs (including associations between input data and desired categories). Hence, this kind of learning process does not meet the definition of neither supervised nor unsupervised learning. Therefore, we call it an **associative learning** which associates neuronal representations of input data or objects defined by various combinations of input data or other internal objects. ANAKG is a special form of hetero-associative memory that can recall an associated data from one category upon presentation of data from another category. The associations are represented by connections between neurons representing various features or objects. ASN neurons adapt using only the input stimuli and internal network processes. When the input stimulus representing the element E_k^n comes, the ASN neuron representing this element in the ANAKG network should be internally activated about this time from other ANAKG neurons which define its context coming from the previous elements of its training sequence. Thus, the input stimulus either confirms the correctness of the neuron activation or starts a tuning process of its weights when the internal activation has come premature or has not occurred at all. If the neuron was recently activated by the time when the input stimulus came, it does not need to update its weights. On the other hand, when this neuron was not activated by the time when the input stimulus came, it means that its stimulations were too weak and synapses should be strengthened. When internal stimuli activate the neuron, but the input signal does not follow it, this activation is undesired. Therefore, the internal connections were too strong and should be weakened to avoid this undesired activation.

The associative tuning process is a new part of an associative training algorithm (ATA) used for the development of ANAKG networks by creating and connecting associative neurons properly and adjusting their weights. In the first phase of the ATA, synaptic weights (4) are initially computed on the basis of synaptic efficacies (1) and permeabilities (2) or (3). During the tuning process, weights are adjusted to remove conflicts and correctly activate subsequent neurons taking into account the context of the previously activated neurons. The associative tuning process must precisely adjust the weights in order to avoid undesired activations of neurons which should not be activated in each given training context defined by training sequences. This tuning process should also avoid premature activations of neurons which can be a result of incomplete contexts of the previous stimulations by their predecessors. Such premature activation can yield incorrect results or damage the subsequent recalling of associated elements. Hence, the synaptic weights must be precisely adjusted. Moreover, when the ANAKG deals with similar training sequences which start with the same initial elements (e.g. words), it is impossible to distinguish them, and these training sequences compete about the same context to activate different neurons. Such multi-activation must be avoided, and a richer unique context is required. Synapses cannot also be too weak because neurons will not spike at all and the recalling process of the next subsequent elements will be totally stopped. At the beginning of the sentences, the contexts for the next element can be sometimes ambiguous, and training sequences compete for strengthening weights of their successors, making the adaptation process unstable. In order to correctly adapt ANAKG networks, the associative tuning process must satisfy all these conditions as far as possible.

In this paper, to deal with strongly correlated training sequences, we developed operations for weakening and strengthening of multiplication factors m in (4). The **weakening operation** which appropriately decreases multiplication factors of active synapses will be used in the cases when the input stimulations of a given neuron were too strong, activating it incorrectly for a given context, or prematurely, i.e. in the reduced input context. The **strengthening operation** will be used when an appropriate neuron was not activated in the full input context, i.e. the input stimulations were too weak.

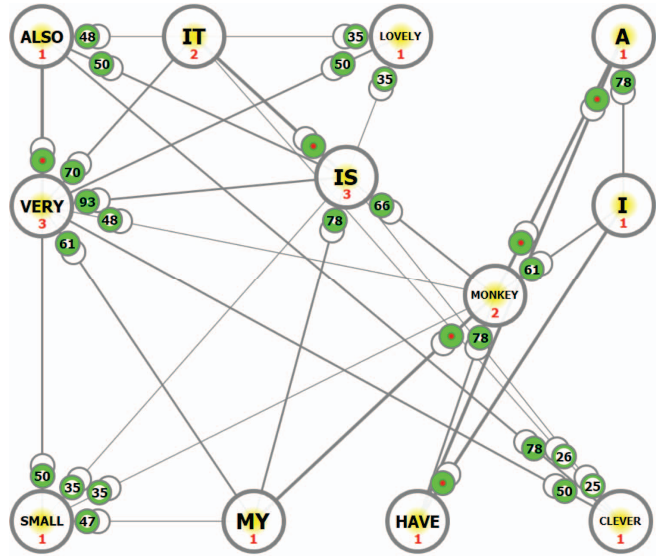

Fig. 2. The untuned ANAKG network created for the four training sequences having trouble with recalling the last very correlated training sequence "It is also very clever" correctly.

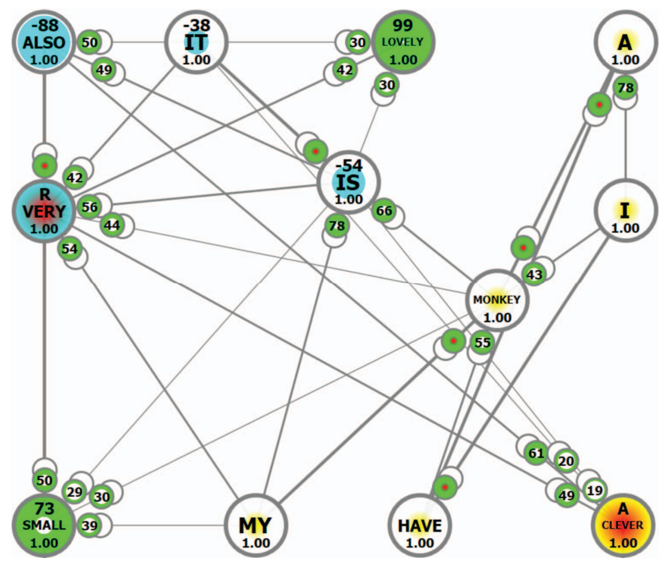

Fig. 3. The fine-tuned ANAKG network (after a single tuning cycle) stimulated by the initial context "It is also" now activates the subsequent neurons "very clever" correctly as defined by the 4th training sequence. The network is presented during its work: the green neurons are charging or relaxing, the cyan neurons are in their relative refraction periods, the cyan-red neuron is in the absolute refraction period, the yellow-red neuron has been just activated and start spiking, and the white-yellow neurons are in their resting states. The presented network recalls all training sequences correctly for all the minimum unique initial contexts, i.e.: "I" → "I have a monkey", "My" → "My monkey is very small", "It is very" → "It is very lovely", and "It is also" → "It is also very clever". The differences in the connection weights can be observed here when comparing the fine-tuned weights here to the weights presented in Fig. 2.

In the small ANAKG network shown in Fig. 2 that was created using the algorithm presented in [7] for the following sequences: "*I have a monkey. My monkey is very small.* ***It is very*** *lovely.* ***It is*** *also* ***very*** *clever.*" This network correctly recalls almost all training sequences, e.g. "*I have a monkey*" for the initial stimulation by "*I*", or "*My monkey is very small*" for the initial stimulation by "*My*" and "*It is very lovely*" for the initial stimulation by "It", with the exception of the last training sequence where it could not differentiate between very similar contexts "*It is very*" and "*It is also*".

The tuning algorithm presented in this section adjusts the synaptic weights as shown in Fig. 3 and achieves the correct subsequent activations of all neurons representing following words in all training sequences for all initial unique contexts, especially for the shortest ones.

Training sequences can compete about appropriate changes of synapses for their initial contexts. These contexts can be exactly the same for several sequences (e.g. "*It is*" is the same for the 3rd and 4th training sequences from the above-mentioned short sample), so it is necessary to avoid destruction and flickering of weights which will be adapted alternately by the currently trained sequence. During the fine-tuning operation, only the multiplication factors *m* are modified without any changes to previously computed permeability factors *p*.

Weights Tuning Algorithm:

The tuning algorithm is divided into two parts which are repeated starting from the first step and finishing at the same step until all training sequences are recalled correctly for all unique initial contexts or it is detected that some training sequences cannot be perfectly recalled:

1. All undesired and premature activations of neurons are avoided for all training sequences by using weakening operations described later in this section. It is necessary to remove all such incorrect activations first because they can impact other neurons and their subsequent activations.
2. Conflicts between correlated training sequences are fine-tuned using strengthening operations described later in this section. The strengthening operations are always followed by the weakening operations (repeating Step 1) for all training sequences in order to maintain the network activity without undesired and premature activations.

Details of these two steps are explained in what follows.

In order to correctly strengthen or weaken weights during the fine-tuning process, it is necessary to compute a maximum dynamic charge level x_{all}^{max} of each considered neuron, starting from the initial value $x_{all}^{max} = 0$ after the absolute refraction of this neuron. The **maximum dynamic charge level** x_{all}^{max} describes a possible maximum level of a neuronal state when the charging for the current stimulus s_{last}^{charge} will be finished. This level is updated always after new stimulus is received according to the following condition:

$$x_{all}^{max} = \begin{cases} x + s_{last}^{charge} & if\ x + s_{last}^{charge} > x_{all}^{max} \\ x_{all}^{max} & otherwise \end{cases} \quad (5)$$

The previously established maximum charge level before adding a new stimulus is always stored in $x_{context}^{max}$ (6). $x_{context}^{max}$ is used to avoid premature activations in the reduced contexts, i.e. activating a neuron without stimulating it by the neurons representing all predecessors of the currently trained sequence.

$$x_{context}^{max} = x_{all}^{max} - s_{last}^{charge} \quad (6)$$

The correct activation of the neuron assumes that only the last charging stimulus made by the direct predecessor of the element in the training sequence can lift the maximum charging level x_{all}^{max} over the activation threshold θ of this neuron

$$x_{context}^{max} < \theta \leq x_{context}^{max} + s_{last}^{charge}. \quad (7)$$

It does not matter how strong this lift (the stimulus s_{last}^{charge}) will be only if the condition (7) is true. On the other hand, i.e. when $x_{context}^{max} \geq \theta$, it means that the activation (spike) occurs prematurely, i.e. for a reduced context of the previously activated neurons.

On the other hand, when $x_{all}^{max} < \theta$, it means that the neuron will not be activated (as desired) even for the full context represented by the stimuli coming from the neurons representing all preceding elements of the currently trained sequence. Therefore, the fine-tuning process aims to satisfy the condition $x_{context}^{max} < \theta \leq x_{all}^{max}$ for the stimulations of all correctly activated predecessors recursively for all subsequent neurons of each training sequence when the input contexts of various training sequences are unique. If any other unintended neuron is activated by the currently trained sequence it can cause extra, unintended stimulations or even activations of other neurons in the network. Therefore, during the tuning algorithm, we assume that all predecessors stimulating the currently adjusted neuron were already fine-tuned and stimulate the currently adjusted neuron correctly.

A. Weakening Operation

The weakening operation defines the way how the multiplication factors *m* of activated neurons are decreased in cases when these neurons were activated in wrong contexts or prematurely in the reduced contexts. To achieve this goal, the multiplication factors (8) are proportionally decreased accordingly to the strengths of synaptic stimulations that made the unwanted activation of this neuron:

$$m = m \cdot \gamma \quad (8)$$

where

$$\gamma = \begin{cases} \frac{\theta}{(x_{all}^{max} + \varepsilon)} & for\ the\ undesired\ activations \\ \frac{\theta}{(x_{context}^{max} + \varepsilon)} & for\ the\ premature\ activations \end{cases} \quad (9)$$

The decrease of the involved weights is made in such a way that the updated weights should stimulate a neuron slightly under its activation threshold θ controlled by a small factor ε, here $\varepsilon = 0.01$. We can distinguish two situations that must be avoided:

1. For undesired activations, the maximum dynamic charge level x_{all}^{max} of the incorrectly activated neuron must be decreased slightly under the threshold ($x_{all}^{max} < \theta$) in the context which resulted in the wrong activation (9).
2. For premature activations, the maximum dynamic charge level $x_{context}^{max}$ of the previously activated neurons must be decreased slightly under the activation threshold ($x_{context}^{max} < \theta$) in this context (9).

B. Strengthening Operation

The strengthening operation defines the way how the multiplication factors m (8) of the insufficiently stimulated neurons are strengthened in cases when these neurons should be activated for the given contexts, but they were not. The increase of multiplication factors of the involved synapses is also conducted proportionally to their stimulation strengths. Sometimes, synapses cannot be strengthened to avoid instability of the tuning process, when the initial context of the tuned neuron is not unique for different training sequences. As a result, the associative recalling of subsequent neurons will be stopped in cases when initial non-unique subsequence does not allow to choose which next subsequent neuron should be recalled.

Synaptic weights are too small for neurons that should be activated when $0 < x_{all}^{max} < \theta$, so the synaptic multiplication factors m of the synapses (8), which took part in the stimulation of this neuron, must be strengthened to satisfy the condition $x_{all}^{max} \geq \theta$. The last stimulus s_{last}^{chagre} can be of any strength which ensures that $x_{context}^{max} < \theta \leq x_{context}^{max} + s_{last}^{chagre}$ will be true. The strengthening of the multiplication factor (4) is computed to allow the neuron achieving its activation threshold for the currently trained sequence in the following way:

$$\gamma = \frac{\theta}{x_{all}^{max} - \varepsilon} \tag{10}$$

IV. Comparison and Evaluation of Results

In order to evaluate and compare solutions without and with the presented tuning algorithm, the quantitative evaluation criterion was defined on the basis of the correctly activated neurons for all unique contexts of all training sequences and the number of all elements in the unique contexts of all training sequences:

$$\underset{S^n \in \mathbb{S}}{\text{cor}} = \frac{\#\{correctly\ acttivated\ neurons\ in\ unique\ contexts\}}{\#\{all\ elements\ in\ unique\ contexts\}} \tag{11}$$

Similarly, a measure of incorrect activations of neurons is:

$$\underset{S^n \in \mathbb{S}}{\text{incor}} = 1 - \underset{S^n \in \mathbb{S}}{\text{cor}} \tag{12}$$

These criteria will be used to evaluate simulation results for various training sequence datasets comparing the results for the untuned and fine-tuned ANAKG networks.

In the sample experiments presented in this section, 4 sequences (introduced in the previous section), 25 sequences (presented in this section), and hundreds of training sequences from Grimm's Fairy Tales were used to evaluate and compare results using the defined evaluation criteria (11) and (12). The 25 very correlated training sequence set consisted of the following sequences: *"I have a monkey. My monkey is very small. It is very lovely. It likes to sit on my head. It can jump very quickly. It is also very clever. It learns quickly. My monkey is lovely. I also have a big cat. My son also has a monkey. It likes to sit on his lamp. I have an old sister. She is very lovely. My sister has a small cat. She likes to sit in the library and read books. She quickly learns languages. My sister has a cat. It is very small. You have a cat as well. It is big. I have a young brother. My brother is small. He has a monkey and dogs. His monkey is small as well. We have lovely dogs."* was used.

TABLE 1. THE COMPARISON OF RESULTS OF THE UNTUNED AND FINE-TUNED ANAKG NETWORKS.

Input Stimulations	ANAKG Responses	Tuned ANAKG Responses
I also	I also have a	I also have a monkey
I have	I have a	I have a
I have an	I have a an old sister	I have an old sister
I have a young	I have a young brother	I have a young brother
I also have a big	I also have a big cat	I also have a big cat
You	You have a cat	You have a cat as well
My son	My son also has a monkey	My son also has a monkey
My brother	My brother	My brother is small
My monkey	My monkey	My monkey is
My monkey is very	My monkey is very small	My monkey is very small
It can	It can jump very quickly	It can jump very quickly
It is also	It is also very	It is also very clever
It is very	It is very	It is very lovely
It learns	It learns	It learns quickly
It likes to	It likes to sit on my his lamp monkey is small	It likes to sit on his lamp
She is very	She is very	She is very lovely
She likes to	She likes to sit on my his lamp monkey is small	She likes to sit in the library and read books
He	He has a monkey	He has a monkey
He has	He has a monkey	He has a monkey and dogs
His monkey is	His monkey is	His monkey is small
His monkey is small	His monkey is small	His monkey is small as well
We have lovely	We have lovely	We have lovely dogs

Table 1 presents a comparison of results achieved for the ANAKG networks without and with the introduced tuning algorithm for sample input stimulations (input contexts) for recalling correlated sequences of associated words (elements). The untuned ANAKG network for correlated training sequences has a serious problem to recall whole training sequences, while the fine-tuned ANAKG usually answers correctly. The untuned ANAKG network sometimes incorrectly recalls correlated sequences which subsequences are combined incorrectly, while the fine-tuned ANAKG usually chooses one of the training

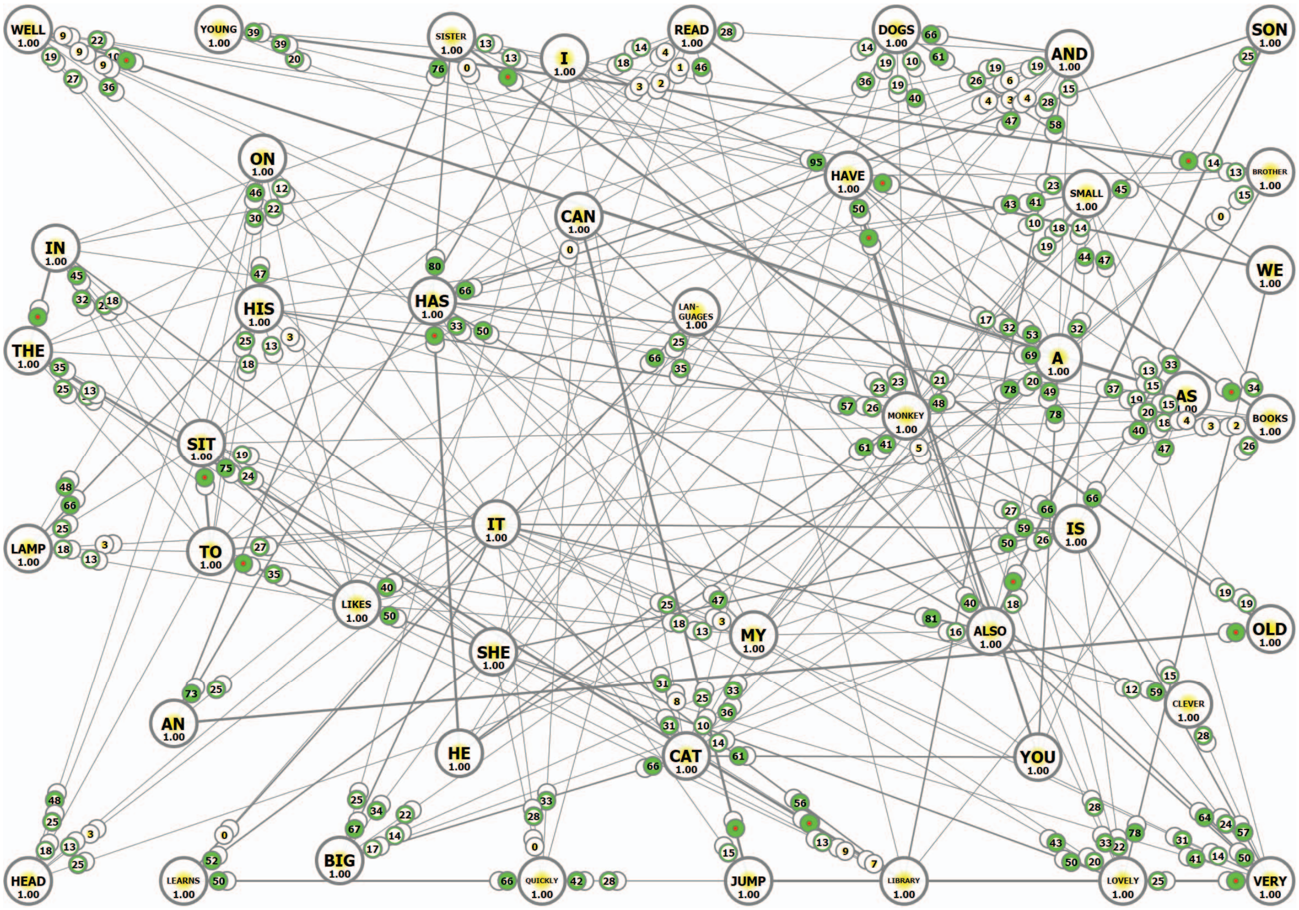

Fig. 4. The tuned ANAKG network created for the 25 training sequences mentioned in the section used for experiments and comparisons of results.

sequences correctly. There are also presented cases when the given input context is ambiguous and insufficient to recall a single training sequence, e.g. "*My monkey*".

Furthermore, two examples of the nastily correlated sequences were considered to describe the way of work of the associative tuning algorithm:

Example 1. If two or more training sequences are correlated, e.g. *"**I have a** monkey. **I also have a** big cat. **I have an** old sister. **I have a** young brother."* it is difficult to appropriately activate the neurons representing the following elements *"monkey, big cat, old sister, or young brother"* respectively. Incorrectly adjusted weights may result in activation of many followers or no one at all. The only distinguishing element is the additional word like *"also"* or *"a/an"*, or the lack of them.

Table 1 presents the associative responses recalled by the ANAKG neural networks constructed for the 25 correlated training sequences listed in this section. The first column contains the selected initial contexts, i.e. subsequences of initial words, which were used to stimulate the ANAKG networks after training. The second column shows the results collected from the ANAKG network [7] without a tuning algorithm, whereas the third column presents the results achieved for the ANAKG network with the implemented tuning. As we can notice, the ANAKG network without tuning sometimes adds incorrect words like "*a*", combines various correlated sequences, presents words in the incorrect order like "*sit on my his lamp monkey*", answers with incorrect followers like "*She likes to sit on my his lamp*", or shortens the recalled sentences, e.g. for "*It is also very*" or „*We have lovely*". The ANAKG network (Fig. 4) fine-tuned with the implemented tuning algorithm responds far more adequately taking into account the listed training sequences.

Example 2. When two or more training sequences begin exactly with the same introductory subsequence (input context), e.g. *"I have a"* in the training sequences *"I have a monkey. I have a young brother."* then it is impossible to distinguish between them at all in an ambiguous context "*I have a*" as presented in Table 1. In such cases, the ANAKG network must be able to stop activations of subsequent neurons when the context is ambiguous.

Table 2 presents the results collected for three sets of correlated training sequences of various sizes. Two of them are presented in this paper, and the third one consists of a three hundreds of selected sentences from Grimms' Fairy Tales. According to the introduced evaluation factors, the correct and incorrect percentage of activated neurons were counted to show the difference between the untuned and fine-tuned ANAKG networks. The achieved results demonstrate that the presented tuning process is very useful especially when training very correlated training sequences, but the results are always better even if the sequences are less correlated than those presented in this paper.

TABLE 2. COMPARISONS OF THE RESULTS FOR THE CONSIDERED TRAINING SEQUENCE SETS USING DEFINED EVALUATION CRITERIA (11) AND (12).

ANAKG:	**UNTUNED**		**FINE-TUNED**	
Evaluation Results:	**cor %**	**incor %**	**cor %**	**incor %**
4 sequences	85%	15%	100%	0%
25 sequences	76%	24%	95%	5%
hundreds of very correlated sequences	54%	46%	91%	9%

The results also confirmed that the tuning process cannot always be perfectly made because the context for each separate training sequence is always limited. As a result, it is impossible to distinguish between some sequences or some parts of sequences which are used in many sequences. Therefore, it would be necessary to use a longer context, e.g. taking into account words from previous sequences or an extra knowledge, to recall desired sequences correctly even if their initial elements are the same. Such processes are typical for people who take into account many factors from the past and their knowledge when talking, thinking, and constructing sentences on the fly. However, this paper tried to show how contextual connections and their fine-tuning can influence the recalling process of associated sequences stored in neuronal structures constructed from associative spiking neurons.

V. CONCLUSIONS

This paper has introduced a new associative tuning algorithm developed for fine-tuning of the ANAKG spiking neural networks [7], [10] devoted to consolidating the representation of training sequences and constructing hetero-associative memories. These neural networks create complex graph structures of associative spiking neurons (ASN), receptors, and effectors to represent training sequences with the aggregated neuronal representations of frequent elements. The main contribution of this paper was to present the associative algorithm for fine-tuning of synaptic weights. The synaptic weights have been defined as a product of the three factors: behavior, permeability, and multiplication. The permeability was determined on the basis of the previously defined synaptic efficacy introduced in [7] and [10]. The multiplication factors were introduced here to create fine-tuned associations which allow for more precise recalling of training sequences from ANAKG networks. The presented tuning algorithm is especially important when we deal with correlated and difficult to distinguish training sequences due to the very similar initial sequence elements or identical elements occurring in different contexts. The presented ANAKG networks together with the implemented tuning of weights allow to use them as semantic memories for storing and contextually recalling various training sequences. The conducted experiments confirmed the effectiveness of the introduced tuning algorithm.

ACKNOWLEDGMENT

This work was supported by a grant from the National Science Centre of Poland DEC-2016/21/B/ST7/02220 and by AGH 11.11.120.612.

REFERENCES

[1] S. L. Bressler and V. Menon,"Large-scale brain networks in cognition: Emerging methods and principles, Trends Cogn Sci., Vol. 14, No. 6, 2010, pp. 277–290.

[2] W. Duch. Brain-inspired conscious computing architecture. Journal of Mind and Behaviour, Vol. 26, 2005, pp. 1–22.

[3] W. Gerstner and W. Kistler. Spiking Neuron Models Cambridge University Press, 2002.

[4] I. Goodfellow, Y. Bengio, A. Courville. Deep Learning, MIT Press, 2016.

[5] D. Graupe. Deep Learning Neural Networks. World Scientific, 2016.

[6] A. Horzyk. Artificial Associative Systems and Associative Artificial Intelligence, EXIT, Warsaw, 2013.

[7] A. Horzyk, J.A. Starzyk, J. Graham. Integration of Semantic and Episodic Memories. IEEE Transactions on Neural Networks and Learning Systems, IEEE Press, Vol. 28, Issue 12, 2017, pp. 3084–3095, DOI: 10.1109/TNNLS.2017.2728203.

[8] A. Horzyk, J.A. Starzyk. Fast Neural Network Adaptation with Associative Pulsing Neurons, In: 2017 IEEE Symposium Series on Computational Intelligence, IEEE Xplore, 2017, pp. 339-346, 2017, DOI: 10.1109/SSCI.2017.8285369.

[9] A. Horzyk, J.A. Starzyk, and Basawaraj, Emergent creativity in declarative memories, 2016 IEEE Symposium Series on Computational Intelligence in Greece, 2016, pp. 1-8, DOI: 10.1109/SSCI.2016.7850029.

[10] A. Horzyk. How Does Generalization and Creativity Come into Being in Neural Associative Systems and How Does It Form Human-Like Knowledge? Elsevier, Neurocomputing, Vol. 144, 2014, pp. 238–257.

[11] E.M. Izhikevich. Simple Model of Spiking Neurons. IEEE Transactions on Neural Networks, Vol. 14, No. 6, 2003, pp. 1569–1572.

[12] J.W. Kalat. Biological grounds of psychology. 10th Ed., Wadsworth Publishing, 2008.

[13] W. Maass. Networks of spiking neurons: The third generation of neural network models, Neural Networks, Vol. 10, Issue 9, Elsevier, 1997, pp. 1659-1671.

[14] J.A. Starzyk, J. Graham. MLECOG - Motivated Learning Embodied Cognitive Architecture. IEEE Systems Journal, Vol. 11, No. 3, 2017, pp. 1272-1283.

[15] L. F. Barrett and A. B. Satpute, Large-scale brain networks in affective and social neuroscience: Towards an integrative functional architecture of the brain, Current Opinion Neurobiology, Vol. 23, No. 3, 2013, pp. 361–372.

[16] R. Tadeusiewicz. New Trends in Neurocybernetics. Computer Methods in Materials Science, Vol. 10, No. 1, 2010, pp. 1–7.

[17] V.A. Nguyen, J.A. Starzyk, W.-B. Goh, and D. Jachyra, Neural Network Structure for Spatio-Temporal Long-Term Memory, IEEE Transactions on Neural Networks and Learning Systems, IEEE Press, Vol. 23, Issue 6, 2012, pp. 971–983.

[18] J.A. Starzyk, Heibo He, Spatio–Temporal Memories for Machine Learning: A Long-Term Memory Organization, IEEE Transactions on Neural Networks and Learning Systems, . Vol. 20, Issue 5, 2009, pp. 768–780.